ARTIFICIAL INTELLIGENCE

(CSC 462)

LAB ASSIGNMENT # 3

NAME: MUAAZ BIN MUKHTAR

REG NO: FA21-BSE-045

CLASS & SECTION: BSSE-5A

SUBMITTED TO: SIR WAQAS ALI

DATE SUBMITTED: 23-12-2023

Department of Computer Science

QUESTION 1

In a four queens problem, a board is a four-by-four grid of squares. A queen is a chess piece that can move on the chessboard any number of squares along any row, column, or diagonal. A queen is attacking another piece if, in a single move, it can move to the square the piece is on without jumping over any other piece. If the other piece is in the line of sight of the queen, then it's attacked by it). The four queens problem poses the question of how four queens can be placed on a chessboard without any queen attacking another queen.

Answer:

Code:

```
def print_solution(board):
    for row in board:
        print(" ".join(row))



def is_safe(board, row, col, n):
    for i in range(row):
        if board[i][col] == 'Q':
            return False

    for i, j in zip(range(row, -1, -1), range(col, -1, -1)):
        if board[i][j] == 'Q':
            return False

    for i, j in zip(range(row, -1, -1), range(col, n)):
        if board[i][j] == 'Q':
            return False

    return True
```

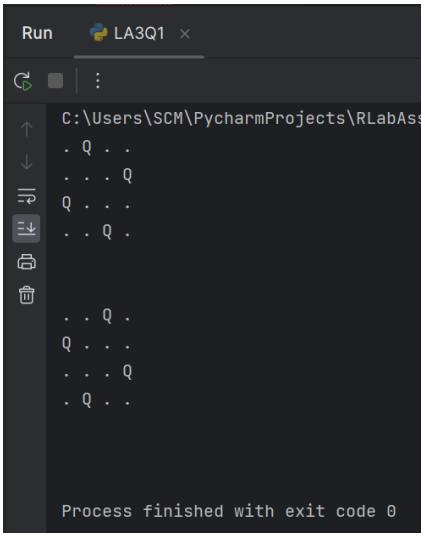
```python
def solve_n_queens_util(board, row, n):
    if row == n:
        print_solution(board)
        print("\n")
        return

    for col in range(n):
        if is_safe(board, row, col, n):
            board[row][col] = 'Q'
            solve_n_queens_util(board, row + 1, n)
            board[row][col] = '.'


def solve_n_queens(n):
    board = [['.' for _ in range(n)] for _ in range(n)]
    solve_n_queens_util(board, 0, n)


solve_n_queens(4)
```

Output:



Question No. 2:

Create a knowledge base which defines your family tree and make a query that uses application of modus ponens to derive a fact which is not explicitly elaborated in the knowledge base.

Answer:

Output:

```
male(john).
male(alex).
male(mike).
female(mary).
female(anna).

 parent(john, mike).
parent(mary, mike).
parent(john, anna).
parent(mary, anna).
parent(mike, alex).

 brother(X, Y) :-
    parent(Z, X),
    parent(Z, Y),
    male(X),
    X \= Y.

 uncle(X, Y) :-
    parent(Z, Y),
    brother(X, Z).

 ancestor(X, Y) :-
    parent(X, Y).
ancestor(X, Y) :-
    parent(X, Z),
    ancestor(Z, Y).
```